

# **Institute of Space Technology Islamabad**

## **Programming Fundamental**

### **Assignment 01**



#### **Submitted to**

Ma'am Tayyiba Arshad

Ma'am Reeda Saeed

#### **Submitted by**

Muhammad Bin Nasir

# LAB TASKS

(Task 1 was submitted)

## Task 2:

Define and initialize five arrays of integer types each having 6 elements and an array of pointers p of size 5. Store the starting address of each array to an array of pointers p. Design and call the function and initialize the values of these five arrays randomly using for loop and p inside this function. we can consider pas a 2D matrix. However, in that case, our number of rows (Which were 5 as there were 5 arrays stored in p) and the number of columns (which were 6 as each array had 6 elements) were fixed. In this question, your goal is to sort all data in the 2d array.

Code:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int temp = 0,temp2;
7      int **arr = new int *[5];
8      cout<<"Give values of arrays.";
9      for (int i=0;i<5;i++)
10     {
11         arr[i] = new int [6];
12     }
13     for (int i=0;i<5;i++)
14     {
15         for (int j=0;j<6;j++)
16         {
17             cin>>*(arr+i)+j);
18         }
19     }
20
21     for (int initial=0;initial<25;initial++)
22     {
23         for (int i=0;i<5;i++)
24         {
25             for (int j=0;j<6;j++)
26             {
27                 if (j<5)
28                 {
29                     if (arr[i][j]>arr[i][j+1])
30                     {
```

```

29         if (arr[i][j]>arr[i][j+1])
30         {
31             temp = arr[i][j];
32             arr[i][j] = arr[i][j+1];
33             arr[i][j+1] = temp;
34         }
35     }
36 }
37 else if (j==5 && i!=4)
38 {
39     if (arr[i][j]>arr[i+1][0])
40     {
41         temp = arr[i][j];
42         arr[i][j] = arr[i+1][0];
43         arr[i+1][0] = temp;
44     }
45 }
46 }
47 }
48 }
49
50 for (int i = 0; i<5; i++)
51 {
52     for (int j=0; j<6; j++)
53     {
54         cout<<arr[i][j]<<" ";
55     }
56     cout<<endl;
57 }
58
59 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP lab\assignment 4\" ; if ($?) { g++ 220201012_Lab4_Problem02.cpp
} { .\220201012_Lab4_Problem02 }
Give values of arrays.3
6
1
3
4
2
6
9
7
6
5
7
8
5
3
6
2
4
6
8
7
6
5
4
2
8
6
8
8
9

```

```
{
{ 1 2 2 2 3 3 }
{ 3 4 4 4 5 5 }
{ 5 6 6 6 6 6 }
{ 6 6 7 7 7 8 }
{ 8 8 8 8 9 9 }
}
```

```
PS D:\University\Assignments\Semester 2\OOP lab\assignment 4> █
```

## Task 3:

In this task, we will extend the previous task and make our number of columns variables. To do this, instead of using arrays of a fixed size of 6 elements, we will make each array to have `ncol` (passed as an argument to the function) elements and store them on heap using `new` operator. Once again store the starting address of each array to an array of pointers `p`. Next, initialize the values of these five arrays randomly using `for` loop and `p`. In this question, your goal is to use the created 2D array where columns represent marks of students and rows represent sections of the OOP course. As you can observe by looking at each of the sections marks there is a huge marking difference, As a C++ developer, you're required to design a function that will apply the scaling to all the marks between [min:50-max:80] and get all the sections marks on the same scale

## Code:

```
1  #include <iostream>
2  using namespace std;
3
4  int scaling(float num, float maximum, float minimum)
5  {
6      double x_std;
7      x_std = (num - minimum)/(maximum-minimum);
8      int x_scaled = (x_std*30) + 50;
9      return x_scaled;
10 }
11
```

```

12  ∨ int main()
13  {
14      float maximum=0,minimum=99999;
15      int studentNumber,temp=0;
16      int **marks = new int *[5];
17      int arr[5];
18  ∨  for (int i=0;i<5;i++)
19      {
20          cout<<"How many students in section "<<i+1<<endl;
21          cin>>studentNumber;
22          marks[i] = new int [studentNumber];
23          arr[i] = studentNumber;
24          cout<<"Give the marks of students of section "<<i+1<<endl;
25  ∨  for (int j=0;j<studentNumber;j++)
26      {
27          cin>>marks[i][j];
28      }
29  ∨  for (int j=0;j<studentNumber-1;j++)
30      {
31  ∨      if (marks[i][j]<marks[i][j+1] && marks[i][j]<minimum)
32          {
33              minimum = marks[i][j];
34          }
35  ∨      else if (marks[i][j+1]<marks[i][j] && marks[i][j+1]<minimum)
36          {
37              minimum = marks[i][j+1];
38          }
39  ∨      if (marks[i][j]>marks[i][j+1] && marks[i][j]>maximum)
40          {
41              maximum = marks[i][j];
42

```

```

42      }
43      else if(marks[i][j+1]>marks[i][j] && marks[i][j+1]>maximum)
44      {
45          maximum = marks[i][j+1];
46      }
47  }
48
49  for (int j=0;j<studentNumber;j++)
50  {
51      marks[i][j] = scaling(marks[i][j],maximum,minimum);
52  }
53  }
54
55
56  for (int i=0;i<5;i++)
57  {
58      for (int j=0;j<arr[i];j++)
59      {
60          cout<<marks[i][j]<<" ";
61      }
62      cout<<endl;
63  }
64
65  }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP lab\assignment 4\" ; if ($?) { g++ 220201012_Lab4_Problem03.cpp
} { .\220201012_Lab4_Problem03 }
How many students in section 1
3
Give the marks of students of section 1
34
43
52
How many students in section 2
4
Give the marks of students of section 2
21
34
32
35
How many students in section 3
5
Give the marks of students of section 3
22
54
67
89
13
How many students in section 4
3
Give the marks of students of section 4
45
65
78
How many students in section 5
2
Give the marks of students of section 5
45
67

50 65 80
50 62 60 63
53 66 71 80 50
62 70 75
62 71
PS D:\University\Assignments\Semester 2\OOP lab\assignment 4>

```

## Task 4:

Part A: Now you are quite comfortable with 2D pointers. Here your goal is to define and allocate memory for 3D pointers. Write a function that receives four arguments:

- An alias to a 3D pointer

- Number of pages (or number of matrices)

- Number of rows

- Number of columns

Now your goal is to first allocate the memory for pages, rows and then for columns dynamically using a new operator. After creating a 3D array now assign the values to each pixel(index) of the 3D array Randomly consider this 3D array as Red, Green, Blue (RGB) Image, Calculate the mean and Stander deviation of RGB image and return the mean and standard deviation of the pixel.

Part B: Write one more function in this function, your goal is to write code for deallocating a dynamically allocated 3D matrix. Your function will receive four arguments:

- An Alias to 3D pointer

- Number of pages

- Number of rows
- Number of columns

Complete the code to properly deallocate the 3D array

Code:

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int allocate3DArray(float ***array, int pages, int rows, int columns)
6  {
7      cout<<"Give the RGB values of pixels"<<endl;
8      array = new float**[pages];
9      for (int i=0;i<pages;i++)
10     {
11         array[i] = new float*[rows];
12         for (int j=0;j<rows;j++)
13         {
14             array[i][j] = new float[columns];
15             for (int k=0;k<columns;k++)
16             {
17                 cin>>array[i][j][k];
18             }
19         }
20     }
21     return 0;
22 }
23
24 int calculateMeanAndStdDev(float ***array, int pages, int rows, int columns, float &mean, float &standardDeviation)
25 {
26     float sum = 0.0,sumSquares = 0.0;
27     int count=0;
28     for (int i=0;i<pages;i++)
29     {
30         for (int j=0;j<rows;j++)
31         {
32             for (int k=0;k<columns;k++)
33             {
34                 float pixel = array[i][j][k];
35                 sum += pixel;
36                 sumSquares += pixel * pixel;
37                 count += 1;
38             }
39         }
40     }
41     mean = sum / count;
42     standardDeviation = sqrt((sumSquares / count) - (mean * mean));
43     return 0;
44 }
45
46 int deallocate3DArray(float ***array, int pages, int rows, int columns)
47 {
48     for (int i = 0;i<pages;i++)
49     {
50         for (int j=0;j<rows;j++)
51         {
52             delete[] array[i][j];
53         }
54         delete[] array[i];
55     }
56     delete[] array;
57     return 0;
58 }

```

```

59  int main()
60  {
61      int pages,rows,columns;
62      float mean,standardDeviation;
63      float ***array;
64
65      cout<<"Give the number of pages,rows and columns of 3D array:"<<endl;
66      cout<<"Pages: ";
67      cin>>pages;
68      cout<<"Rows: ";
69      cin>>rows;
70      cout<<"Columns: ";
71      cin>>columns;
72
73      allocate3DArray(array, pages, rows, columns);
74
75      calculateMeanAndStdDev(array, pages, rows, columns, mean, standardDeviation);
76      cout << "Mean: " << mean << endl;
77      cout << "Standard Deviation: " << standardDeviation << endl;
78
79      deallocate3DArray(array, pages, rows, columns);
80  }

```

## Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP lab\assignment 4\" ; if ($?) { g++ 220201012_Lab4_Problem04.cpp
} { .\220201012_Lab4_Problem04 }
Give the number of pages,rows and columns of 3D array:
Pages: 3
Rows: 4
Columns: 3
Give the RGB values of pixels
23
32
44
22
32
45
43
32
25
65
45
54
43
32
34
23
16
76
76
65
45
54
65
56
76
56
78
54
54
65
56
76
56
78
54
35
44
34
35
53
35
67
65
54
Mean: 46.0278
Standard Deviation: 16.3749
PS D:\University\Assignments\Semester 2\OOP lab\assignment 4> █

```



## Task 5:

Just like in Task 4, we will extend task 4 and make this student's marks scaling problem as departmental level scaling problem where you have multiple sections in each department and a university can have multiple departments, for better understanding consider this as a 3D problem where Sections and Students data will be saved in 2 dimensional and each departments record will be saved in 3<sup>rd</sup> dimension. Design a function that can perform the following steps

- Create a 3D array dynamically
- Assign random marks to each student of different sections of different department
- Apply the min-max scaling to this 3D array
- Display the marks before and after scaling

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  void allocate3DArray(float ***arr, int depts, int sects, int student)
5  {
6      cout<<"Give the marks of stdents:"<<endl;
7      arr = new float**[depts];
8      for (int i=0;i<depts;i++)
9      {
10         cout<<"Students of Department "<<i+1<<":"<<endl;
11         arr[i] = new float*[sects];
12         for (int j=0;j<sects;j++)
13         {
14             cout<<"Students of Section "<<j+1<<":"<<endl;
15             arr[i][j] = new float[student];
16             for (int k=0;k<student;k++)
17             {
18                 cout<<"Marks of student "<<k+1<<":"<<endl;
19                 cin>>arr[i][j][k];
20             }
21         }
22     }
23 }
24
25 int scaling(float num,float maximum,float minimum)
26 {
27     double x_std;
28     x_std = (num - minimum)/(maximum-minimum);
29     int x_scaled = (x_std*30) + 50;
30     return x_scaled;
31 }
32
33 int main()
34 {
35     int depts,sects,students;
36     float maximum=0,minimum=99999;
37     cout<<"Give the number of departments: ";
38     cin>>depts;
39     cout<<"Give the number of sections: ";
40     cin>>sects;
41     cout<<"Give the number of students in each section: ";
42     cin>>students;
43     float ***arr;
44
45     allocate3DArray(arr,depts,sects,students);
46     cout<<endl<<endl;
47     cout<<"The marks after being scaled are:"<<endl;
48     for (int i=0;i<depts;i++)
49     {
50         cout<<endl<<"Students of department "<<i<<":"<<endl;
51         for (int j=0;j<sects;j++)
52         {
53             cout<<"Stdents of section "<<j<<":"<<endl;
54             for (int k=0;k<students;k++)
55             {
56                 cout<<arr[i][j][k]<<" ";
57             }
58             cout<<endl;
59         }
60     }

```

```

61     for (int i=0;i<depts;i++)
62     {
63         for (int j=0;j<sects;j++)
64         {
65             for (int k=0;k<students-1;k++)
66             {
67                 if (arr[i][j][k]<arr[i][j][k+1] && arr[i][j][k]<minimum)
68                 {
69                     minimum = arr[i][j][k];
70                 }
71                 else if (arr[i][j][k+1]<arr[i][j][k] && arr[i][j][k+1]<minimum)
72                 {
73                     minimum = arr[i][j][k+1];
74                 }
75                 if (arr[i][j][k]>arr[i][j][k+1] && arr[i][j][k]>maximum)
76                 {
77                     maximum = arr[i][j][k];
78                 }
79                 else if (arr[i][j][k+1]>arr[i][j][k] && arr[i][j][k+1]>maximum)
80                 {
81                     maximum = arr[i][j][k+1];
82                 }
83             }
84             for (int k=0;k<students;k++)
85             {
86                 arr[i][j][k] = scaling(arr[i][j][k],maximum,minimum);
87             }
88         }
89     }

```

```

92     cout<<endl<<endl;
93     cout<<"The marks after being scaled are:"<<endl;
94     for (int i=0;i<depts;i++)
95     {
96         cout<<endl<<"Students of department "<<i<<":"<<endl;
97         for (int j=0;j<sects;j++)
98         {
99             cout<<"Stdents of section "<<j<<":"<<endl;
100             for (int k=0;k<students;k++)
101             {
102                 cout<<arr[i][j][k]<<" ";
103             }
104             cout<<endl;
105         }
106     }
107
108 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP lab\assignment 4\" ; if ($?) { g++ 220201012_Lab4_Problem05.cpp -
) { .\220201012_Lab4_Problem05 }
Give the number of departments: 3
Give the number of sections: 3
Give the number of students in each section: 3
Give the marks of students:
Students of Department 1:
Students of Section 1:
Marks of student 1:
71
Marks of student 2:
62
Marks of student 3:
91
Students of Section 2:
Marks of student 1:
56
Marks of student 2:
74
Marks of student 3:
43
Students of Section 3:
Marks of student 1:
56
Marks of student 2:
76
Marks of student 3:
34
Students of Department 2:
Students of Section 1:
Marks of student 1:
23
Marks of student 2:
34
Marks of student 3:
42
Students of Section 2:
Marks of student 1:
35
Marks of student 2:
56
Marks of student 3:
67
Students of Section 3:
Marks of student 1:
76
Marks of student 2:
68
Marks of student 3:
67
Students of Department 3:
Students of Section 1:
Marks of student 1:
87
Marks of student 2:
78
Marks of student 3:
89
Students of Section 2:
Marks of student 1:
90
Marks of student 2:

```

```
The marks after being scaled are:
```

```
Students of department 0:
```

```
Stdents of section 0:
```

```
59 50 80
```

```
Stdents of section 1:
```

```
58 69 50
```

```
Stdents of section 2:
```

```
61 72 50
```

```
Students of department 1:
```

```
Stdents of section 0:
```

```
50 54 58
```

```
Stdents of section 1:
```

```
55 64 69
```

```
Stdents of section 2:
```

```
73 69 69
```

```
Students of department 2:
```

```
Stdents of section 0:
```

```
78 74 79
```

```
Stdents of section 1:
```

```
76 80 78
```

```
Stdents of section 2:
```

```
67 71 70
```

```
PS D:\University\Assignments\Semester 2\OOP lab\assignment 4> █
```

```
The marks after being scaled are:
```

```
Students of department 0:
```

```
Stdents of section 0:
```

```
59 50 80
```

```
Stdents of section 1:
```

```
58 69 50
```

```
Stdents of section 2:
```

```
61 72 50
```

```
Students of department 1:
```

```
Stdents of section 0:
```

```
50 54 58
```

```
Stdents of section 1:
```

```
55 64 69
```

```
Stdents of section 2:
```

```
73 69 69
```

```
Students of department 2:
```

```
Stdents of section 0:
```

```
78 74 79
```

```
Stdents of section 1:
```

```
76 80 78
```

```
Stdents of section 2:
```

```
67 71 70
```

```
PS D:\University\Assignments\Semester 2\OOP lab\assignment 4> █
```

## ASSIGNMENT TASKS

### Task 1:

Write a function `secondSmallest()` which takes the two integers A and B respectively, and calculate the possible number whose sum equal to A and digit equal B, afterwards select and return a number that is the second smallest number if possible, else return -1. You have to implement this function using pass by address approach (i.e using pointer as an argument).

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  int secondSmallest(int *sum,int *digits)
5  {
6      int temp=0,remainder,count=0,lowerLimit,upperLimit,result=1;
7      for(int i=0;i<(*digits-1);i++)
8      {
9          result = result * 10;
10     }
11     lowerLimit = result;
12     result = 1;
13     for(int i=0;i<*digits;i++)
14     {
15         result = result * 10;
16     }
17     upperLimit = result-1;
18     for (int i=lowerLimit;i<upperLimit;i++)
19     {
20         int toCheck = i;
21         temp = 0;
22         while(toCheck>0)
23         {
24             remainder=toCheck%10;
25             temp=temp+remainder;
26             toCheck=toCheck/10;
27         }
28         if(temp==(*sum))
29         {
30             count+=1;

```

```

30             count+=1;
31         }
32         if(count==2)
33         {
34             return i;
35             break;
36         }
37     }
38     return -1;
39 }
40
41 int main()
42 {
43     int sum,digits;
44     cout<<"Give the number to which sum should be equal:"<<endl;
45     cin>>sum;
46     cout<<"Give the digits of number:"<<endl;
47     cin>>digits;
48
49     cout<<"The second number whose sum equals "<<sum<<" and "<<digits<<" digits is "<<secondSmallest(&sum,&digits);
50     return 0;
51 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ;
f ($?) { .\220201012_Assign01_Problem01 }
Give the number to which sum should be equal:
6
Give the digits of number:
2
The second number whose sum equals 6 and 2 digits is 24
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ;
f ($?) { .\220201012_Assign01_Problem01 }
Give the number to which sum should be equal:
16
Give the digits of number:
3
The second number whose sum equals 16 and 3 digits is 178
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 2:

Write a function `sumOfElement()` which takes the array (through pointer) and its size as argument, and ask the user to enter a number `S` that is sum of elements present in that particular array. And prompt the user in same function with location of element whose sum is equal to `S`.

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  void sumOfElements(int *arr, int size, int sum, int *first_element, int *second_element)
5  {
6      for(int i=0;i<size;i++)
7      {
8          for(int j=i;j<size;j++)
9          {
10             if ((arr[i]+arr[j])==sum)
11             {
12                 *first_element = i;
13                 *second_element = j;
14                 break;
15             }
16         }
17     }
18 }
19

```



```

20
21 int main()
22 {
23     int first_element,second_element,sum,members;
24     cout<<"How many elements of the array?"<<endl;
25     cin>>members;
26     int arr[members];
27     cout<<"Give the members of array"<<endl;
28     for(int i=0;i<members;i++)
29     {
30         cin>>arr[i];
31     }
32     cout<<"What is the sum?"<<endl;
33     cin>>sum;
34
35     sumOfElements(arr,members,sum,&first_element,&second_element);
36     cout<<"The array given is:"<<endl;
37     cout<<"{ ";
38     for(int i=0;i<members;i++)
39     {
40         cout<<arr[i]<<" ";
41     }
42     cout<<"}"<<endl;
43     cout<<"The elements at index "<<first_element<<" and "<<second_element<<" when added give the sum "<<sum;
44     return 0;
45 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01" ; if ($?) { g++ 220201012_Assign01_Problem02.cpp
f ($?) { .\220201012_Assign01_Problem02 }
How many elements of the array?
4
Give the members of array
34
43
23
43
What is the sum?
66
The array given is:
{ 34 43 23 43 }
The elements at index 2 and 3 when added give the sum 66
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 3:

Write a function of reversearray() that will take array as pointer argument and reverse the element of an array using loop.

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  void reverseArray(int *arr,int size)
5  {
6      int temp;
7      for (int i=0;i<(size/2);i++)
8      {
9          temp = arr[i];
10         arr[i] = arr[size-i-1];
11         arr[size-i-1] = temp;
12     }
13 }
14

```



```

15  int main()
16  {
17      int size;
18      cout<<"What is the size of array?"<<endl;
19      cin>>size;
20      int arr[size];
21      cout<<"Give the members of array:"<<endl;
22      for(int i=0;i<size;i++)
23      {
24          cin>>arr[i];
25      }
26      cout<<"The original array is:"<<endl<<"{ ";
27      for(int i=0;i<size;i++)
28      {
29          cout<<arr[i]<<" ";
30      }
31      cout<<"}"<<endl;
32      reverseArray(arr,size);
33      cout<<"The reversed array is:"<<endl<<"{ ";
34      for(int i=0;i<size;i++)
35      {
36          cout<<arr[i]<<" ";
37      }
38      cout<<"}"<<endl;
39  }
40  }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem03.cpp
f ($?) { .\220201012_Assign01_Problem03 }
What is the size of array?
5
Give the members of array:
45
32
22
21
32
The original array is:
{ 45 32 22 21 32 }
The reversed array is:
{ 32 21 22 32 45 }
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 4:

Write a function of swap () of two numbers using pointer, reflect the swapping of element in main function. (Also you are not required to use temp or any third variable for swapping).

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  void swap(int *num1,int *num2)
5  {
6      *num1 = *num1 + *num2;
7      *num2 = *num1 - *num2;
8      *num1 = *num1 - *num2;
9  }
10
11 int main()
12 {
13     int num1,num2;
14     cout<<"Give the first number:"<<endl;
15     cin>>num1;
16     cout<<"Give the second number:"<<endl;
17     cin>>num2;
18
19     cout<<"Original numbers are:"<<endl;
20     cout<<"Number 1: "<<num1<<endl;
21     cout<<"Number 2: "<<num2<<endl;
22     swap(&num1,&num2);
23     cout<<"Numbers after swapping are:"<<endl;
24     cout<<"Number 1: "<<num1<<endl;
25     cout<<"Number 2: "<<num2<<endl;
26     return 0;
27 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem04.cpp
f ($?) { .\220201012_Assign01_Problem04 }
Give the first number:
45
Give the second number:
67
Original numbers are:
Number 1: 45
Number 2: 67
Numbers after swapping are:
Number 1: 67
Number 2: 45
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 5:

Write a function Agecal() that takes your date of birth in YYYY/MM/DD format (separated by space) as input as well as current date in same format, and calculates your age in years, months and days. You must check for leap years also. Write a separate function to check for leap year. Implement this program using call by pointer approach.

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  bool isLeap(int year)
5  {
6      if (year % 4 != 0)
7      {
8          return false;
9      }
10     else if (year % 100 != 0)
11     {
12         return true;
13     }
14     else if (year % 400 != 0)
15     {
16         return false;
17     }
18     else
19     {
20         return true;
21     }
22 }
23
24 void AgeCal(int *dob,int *cur)
25 {
26     int year,month,day,count=0;
27     cur[1] = cur[1] -1;
28     cur[0] = cur[0] +30;
29     cur[2] = cur[2] -1;
30     cur[1] = cur[1] +12;
31     year = cur[2]-dob[2];
32     month = cur[1]-dob[1];
33     day = cur[0]-dob[0];
34     for(int i=dob[2];i<cur[2];i++)
35     {
36         if (isLeap(i))
37         {
38             count+=1;
39         }
40     }
41     day = day+count;
42     if (day>30)
43     {
44         day = day-30;
45         month = month+1;
46     }
47     if (month>12)
48     {
49         month = month-12;
50         year = year+1;
51     }
52     cout<<"Your age is "<<year<<" years, "<<month<<" months and "<<day<<" days.";
53 }
54
55
56
57
58 int main()
59 {
60     int DateOfBirth[3],CurrentDate[3];
61     cout<<"Give your date of birth (in ddmmyyyy format):"<<endl;
62     cin>>DateOfBirth[0]>>DateOfBirth[1]>>DateOfBirth[2];
63     cout<<"Give the today's date:"<<endl;
64     cin>>CurrentDate[0]>>CurrentDate[1]>>CurrentDate[2];
65
66     AgeCal(DateOfBirth,CurrentDate);
67 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem05.cpp
f ($?) { .\220201012_Assign01_Problem05 }
Give your date of birth (in ddmmyyy format):
19
07
2003
Give the today's date:
20
03
2023
Your age is 19 years, 8 months and 6 days.
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 6:

Write a program to find the factorial of a number using pointer.

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  int factorial(int *ptr)
5  {
6      int fact = 1;
7      for(int i=1;i< *ptr+1;i++)
8      {
9          fact *= i;
10     }
11     return fact;
12 }
13
14 int main()
15 {
16     int num;
17     cout<<"Give a number: "<<endl;
18     cin>>num;
19     cout<<"The factorial of "<<num<<" is: "<<factorial(&num);
20     return 0;
21 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem06.cpp
f ($?) { .\220201012_Assign01_Problem06 }
Give a number:
5
The factorial of 5 is: 120
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 7:

Write a program totake array from user which contains even and odd number, write function to separate out the even and odd element and place even numbers on left side and odd on right side of array.

Code:

```
1  #include <iostream>
2  using namespace std;
3
4  void evenOddSeparator(int *arr,int size)
5  {
6      int tempArray[size];
7      int count=0;
8      for (int i=0;i<size;i++)
9      {
10         if (arr[i]%2==0)
11         {
12             tempArray[count]=arr[i];
13             count+=1;
14         }
15     }
16     for (int i=0;i<size;i++)
17     {
18         if (arr[i]%2!=0)
19         {
20             tempArray[count]=arr[i];
21             count+=1;
22         }
23     }
24     for (int i=0;i<size;i++)
25     {
26         arr[i] = tempArray[i];
27     }
28 }
29
30 int main()
31 {
32     int size;
33     cout<<"Give the size of array: ";
34     cin>>size;
35     int arr[size];
36     cout<<"Give the members of array:"<<endl;
37     for (int i=0;i<size;i++)
38     {
39         cin>>arr[i];
40     }
41     evenOddSeparator(arr,size);
42     cout<<"{ ";
43     for (int i=0;i<size;i++)
44     {
45         cout<<arr[i]<<" ";
46     }
47     cout<<"}";
48
49 }
```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem07.cpp
f ($?) { .\220201012_Assign01_Problem07 }
Give the size of array: 5
Give the members of array:
4
5
6
7
8
{ 4 6 8 5 7 }
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 8:

Write a function which will take pointer and display the number on screen.  
Take number from user and print it on screen using that function in the string.

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  void display(int *num)
5  {
6      cout<<"The number is "<<*num;
7  }
8
9  int main()
10 {
11     int *num = new int;
12     cout<<"Give any number:"<<endl;
13     cin>>*num;
14     display(num);
15 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem08.cpp
f ($?) { .\220201012_Assign01_Problem08 }
Give any number:
55
The number is 55
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 9:

Write a program that take an array of size 10, and split the array into 2 parts,  
and then store the element in two different arrays using pointer approach.

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int *arr = new int[10];
7      int *arr1 = new int[5];
8      int *arr2 = new int[5];
9      cout<<"Give 10 members of array:"<<endl;
10     for (int i=0;i<10;i++)
11     {
12         cin>>arr[i];
13     }
14     for (int i=0;i<5;i++)
15     {
16         arr1[i] = arr[i];
17     }
18     for (int i=0;i<5;i++)
19     {
20         arr2[i] = arr[i+5];
21     }
22     cout<<"New arrays are: "<<endl<<"{ ";
23     for (int i=0;i<5;i++)
24     {
25         cout<<arr1[i]<<" ";
26     }
27     cout<<"}"<<endl;
28     cout<<"{ ";
29     for (int i=0;i<5;i++)
30     {
31         cout<<arr2[i]<<" ";
32     }
33     cout<<"}";
34 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem09.cpp
f ($?) { .\220201012_Assign01_Problem09 }
Give 10 members of array:
2
3
4
5
6
7
8
9
10
11
New arrays are:
{ 2 3 4 5 6 }
{ 7 8 9 10 11 }
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 10:

Given the string "A string". Print on one line the letter on the index 0, the pointer position and letter t. Now update the pointer to pointer+2. then, in another line print the pointer and letters r and gof the string(using the pointer).

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char myString[] = "A string";
7      char *ptr = new char;
8      ptr = myString;
9
10     cout<<*ptr<<" "<<*ptr<<" "<<*(ptr+3)<<endl;
11     ptr = ptr+2;
12     cout<<*ptr<<" "<<*(ptr+2)<<" "<<*(ptr+5)<<endl;
13     return 0;
14 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem10.cpp
f ($?) { .\220201012_Assign01_Problem10 }
A A t
string r g
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 11:

Write a program take the 2D dynamic array of row and columns from user, your program should first shift the element of evenrows (0,2,4.. )to right circularof , and then shift the element of odd columns(1,3,5..)to the left side.

Code:



```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int rows,columns,temp;
7      cout<<"Give the rows and columns of 2D array."<<endl;
8      cout<<"Rows: ";
9      cin>>rows;
10     cout<<"Columns: ";
11     cin>>columns;
12     int **doublePtr = new int*[rows];
13     for (int i=0;i<rows;i++)
14     {
15         doublePtr[i] = new int[columns];
16     }
17     int **newdoublePtr = new int*[rows];
18     for (int i=0;i<rows;i++)
19     {
20         newdoublePtr[i] = new int[columns];
21     }
22     cout<<"Give the elements of array"<<endl;
23     for (int i=0;i<rows;i++)
24     {
25         cout<<"Elements of "<<i+1<<"th row"<<endl;
26         for (int j=0;j<columns;j++)
27         {
28             cin>>doublePtr[i][j];
29         }
30     }
31     for (int i=0;i<rows;i++)
32     {
33         for (int j=0;j<columns;j++)
34         {
35             if(i%2==0)
36             {
37                 if (j!=(columns-1))
38                 {
39                     newdoublePtr[i][j+1] = doublePtr[i][j];
40                 }
41                 else
42                 {
43                     newdoublePtr[i][0] = doublePtr[i][j];
44                 }
45             }
46             if(i%2!=0)
47             {
48                 if (j!=0)
49                 {
50                     newdoublePtr[i][j-1] = doublePtr[i][j];
51                 }
52                 else
53                 {
54                     newdoublePtr[i][columns-1] = doublePtr[i][j];
55                 }
56             }
57         }
58     }
59     cout<<"{ "<<endl;
60     for (int i=0;i<rows;i++)
61     {
62         cout<<"{ ";
63         for (int j=0;j<columns;j++)
64         {
65             cout<<newdoublePtr[i][j]<<" ";
66         }
67         cout<<"}"<<endl;
68     }
69     cout<<"}";
70     return 0;
71 }

```

Output:

```

PS C:\Users\W> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem11.cpp
f ($?) { .\220201012_Assign01_Problem11 }
Give the rows and columns of 2D array.
Rows: 4
Columns: 4
Give the elements of array
Elements of 1th row
1
2
3
4
Elements of 2th row
5
6
7
8
Elements of 3th row
9
0
1
2
Elements of 4th row
3
4
5
6
{
{ 4 1 2 3 }
{ 6 7 8 5 }
{ 2 9 0 1 }
{ 4 5 6 3 }
}
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 12:

write a program that takes an array of length n, where all the numbers are nonnegative and unique. Find the element in the array processing the highest value, split that highest element into two parts where first part contains the next highest value in the array and second part hold the required additive entity to get the highest value. Print the array where the highest value gets splitted into those two parts.

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int size,maximum=0,secondMaximum=0,index,additive;
7      cout<<"Give the size of array:"<<endl;
8      cin>>size;
9      int *array = new int[size];
10     cout<<"Give the elements of array:"<<endl;
11     for (int i=0;i<size;i++)
12     {
13         cin>>array[i];
14     }
15     for (int i=0;i<size;i++)
16     {
17         if (array[i]>maximum)
18         {
19             maximum = array[i];
20             index = i;
21         }
22     }
23     for (int i=0;i<size;i++)
24     {
25         if(array[i]>secondMaximum && array[i]!=maximum)
26         {
27             secondMaximum = array[i];
28         }
29     }
30     additive = maximum - secondMaximum;
31
32     int *newArray = new int[size+1];
33     for (int i=0;i<size;i++)
34     {
35         if (i<index)
36         {
37             newArray[i] = array[i];
38         }
39         if (i==index)
40         {
41             newArray[i] = secondMaximum;
42             newArray[i+1] = additive;
43         }
44         if (i>index)
45         {
46             newArray[i+1] = array[i];
47         }
48     }
49     cout<<"{ ";
50     for (int i=0;i<size+1;i++)
51     {
52         cout<<newArray[i]<<" ";
53     }
54     cout<<"}";
55
56     delete[] array;
57     delete[] newArray;
58     return 0;
59 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem12.cpp
f ($?) { .\220201012_Assign01_Problem12 }
Give the size of array:
5
Give the elements of array:
3
6
4
5
2
{ 3 5 1 4 5 2 }
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 13:

Write a program that takes 2D dynamic array of 3\*3 size. And check if the matrix is symmetric or not. Condition of symmetric is:  $a[i][j] = a[j][i]$

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int count=0;
7      int **ptr = new int*[3];
8      for (int i=0;i<3;i++)
9      {
10         ptr[i] = new int[3];
11     }
12     cout<<"Give the elements of array:"<<endl;
13     for (int i=0;i<3;i++)
14     {
15         for (int j=0;j<3;j++)
16         {
17             cin>>ptr[i][j];
18         }
19     }
20
21     for (int i=0;i<3;i++)
22     {
23         for (int j=0;j<3;j++)
24         {
25             if (ptr[i][j] != ptr[j][i])
26             {
27                 cout<<"The matrix is not symmetric."<<endl;
28                 count +=1;
29                 goto end;
30             }
31         }
32     }
33
34     end:
35     if (count==0)
36     {
37         cout<<"The given matrix is symmetric."<<endl;
38     }
39
40     delete ptr;
41     return 0;

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem13.cpp
f ($?) { .\220201012_Assign01_Problem13 }
Give the elements of array:
2
3
4
5
6
7
8
9
0
The matrix is not symmetric.
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem13.cpp
f ($?) { .\220201012_Assign01_Problem13 }
Give the elements of array:
1
0
0
0
1
0
0
0
1
The given matrix is symmetric.
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>

```

## Task 14:

Write a program that take two 2D dynamic array of size 3\*3, and compute the addition and multiplication.

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int **array1 = new int*[3];
7      for (int i=0;i<3;i++)
8      {
9          array1[i] = new int[3];
10     }
11     cout<<"Give the elements of array:"<<endl;
12     for (int i=0;i<3;i++)
13     {
14         for (int j=0;j<3;j++)
15         {
16             cin>>array1[i][j];
17         }
18     }
19     int **array2 = new int*[3];
20     for (int i=0;i<3;i++)
21     {
22         array2[i] = new int[3];
23     }
24     cout<<"Give the elements of array:"<<endl;
25     for (int i=0;i<3;i++)
26     {
27         for (int j=0;j<3;j++)
28         {
29             cin>>array2[i][j];
30         }
31     }

```

```

33     int **multiplyArray = new int*[3];
34     for (int i=0;i<3;i++)
35     {
36         multiplyArray[i] = new int[3];
37     }
38     int **addArray = new int*[3];
39     for (int i=0;i<3;i++)
40     {
41         multiplyArray[i] = new int[3];
42     }
43
44     for (int i=0;i<3;i++)
45     {
46         for (int j=0;j<3;j++)
47         {
48             multiplyArray[i][j] = array1[i][j] * array2[i][j];
49             addArray[i][j] = array1[i][j] + array2[i][j];
50         }
51     }
52
53     cout<<" ";
54     for (int i=0;i<3;i++)
55     {
56         for (int j=0;j<3;j++)
57         {
58             cout<<addArray[i][j]<<" ";
59         }
60         cout<<endl;
61     }
62     cout<<" " <<endl;
63
64     cout<<" ";
65     for (int i=0;i<3;i++)
66     {
67         for (int j=0;j<3;j++)
68         {
69             cout<<multiplyArray[i][j]<<" ";
70         }
71         cout<<endl;
72     }
73     cout<<" " <<endl;
74
75 }

```

Output:

```

PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem14.cpp
f ($?) { .\220201012_Assign01_Problem14 }
Give the elements of array 1:
1
2
3
4
5
6
7
8
9
Give the elements of array 2:
1
1
1
1
1
1
1
1
1

```

```
Addition gives:
```

```
{  
2 3 4  
5 6 7  
8 9 10  
}
```

```
Multiplication gives:
```

```
{  
1 2 3  
4 5 6  
7 8 9  
}
```

```
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>
```

## Task 15:

Write a program that reads grades of students for multiple assignments, then calculate each student's average grade. The program should use a 2D dynamic array to store the marks, with each row representing an assignment. The program should prompt the user to enter the number of students and the number of assignments, and then allocate the 2d array dynamically based on those values. Prompt the user to enter marks for each student for each assignment. Calculate each student's average marks and display it to the user. Deallocate the 2d dynamic array before exiting the program. Also your program should handle the input validation, such as ensuring that user enters positive integers for number of students and assignment, and the grades entered are valid (i.e. between 0-100).

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int rows,columns,sum=0;
7      assignment:
8      cout<<"How many assignments to store?"<<endl;
9      cin>>int temp;
10     if (temp<0)
11     {
12         cout<<"Invalid entry. Please try again"<<endl;
13         goto assignment;
14     }
15     rows = temp;
16     int **MarksOfStudents = new int*[rows];
17     student:
18     cout<<"How many students are there?"<<endl;
19     cin>>temp;
20     if (temp<0)
21     {
22         cout<<"Invalid entry. Please try again"<<endl;
23         goto student;
24     }
25     columns = temp;
26     for (int i=0;i<rows;i++)
27     {
28         MarksOfStudents[i] = new int[columns];
29     }
30     for (int i=0;i<rows;i++)
31     {
32         cout<<"Assignment " <<i+1<<endl;

```

```

33         for (int j=0;j<columns;j++)
34         {
35             again:
36             cout<<"Give the marks of student " <<j+1<<" ";
37             cin>>int temp;
38             if (temp<0 || temp>100)
39             {
40                 cout<<"Invalid input. Please try again"<<endl;
41                 goto again;
42             }
43             MarksOfStudents[i][j] = temp;
44         }
45     }
46
47     int *AverageMarks = new int[columns];
48     for (int i=0;i<columns;i++)
49     {
50         for (int j=0;j<rows;j++)
51         {
52             sum += MarksOfStudents[j][i];
53         }
54         AverageMarks[i] = sum/columns;
55         sum = 0;
56     }
57
58     cout<<"The average marks of students are:"<<endl;
59     for (int i=0;i<columns;i++)

```

```

60     {
61         cout<<"Student " <<i+1<<" : " <<AverageMarks[i]<<endl;
62     }
63     return 0;
64 }

```



## Output:

```
PS C:\Users\M> cd "d:\University\Assignments\Semester 2\OOP\Assignment 01\" ; if ($?) { g++ 220201012_Assign01_Problem15.cpp
f ($?) { .\220201012_Assign01_Problem15 }
How many assignments to store?
5
How many students are there?
4
Assignment 1
Give the marks of student 1: 11
Give the marks of student 2: 12
Give the marks of student 3: 19
Give the marks of student 4: 15
Assignment 2
Give the marks of student 1: 12
Give the marks of student 2: 14
Give the marks of student 3: 19
Give the marks of student 4: 17
Assignment 3
Give the marks of student 1: 9
Give the marks of student 2: 8
Give the marks of student 3: 17
Give the marks of student 4: 16
Assignment 4
Give the marks of student 1: 14
Give the marks of student 2: 13
Give the marks of student 3: 20
Give the marks of student 4: 17
Assignment 5
Give the marks of student 1: 19
Give the marks of student 2: 14
Give the marks of student 3: 11
Give the marks of student 4: 7

The average marks of students are:
Student 1: 13
Student 2: 12
Student 3: 17
Student 4: 14
PS D:\University\Assignments\Semester 2\OOP\Assignment 01>
```